ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

ON A NASH GAME FOR TOPOLOGY OPTIMIZATION UNDER LOAD-UNCERTAINTY – FINDING THE WORST LOAD

Carl-Johan Thore[1]

[1]Division of Solid Mechanics, Department of Management and Engineering, Linköping University
Linköping University, 581 83 Linköping, Sweden
e-mail: carl-johan.thore@liu.se

Keywords: Worst load, Global optimization, Branch and Bound, Topology optimization, Nash game

Abstract. *Topology optimization (TO) enables creation of structures which are highly efficient with respect to precisely those conditions formulated mathematically in the optimization problem. To ensure that optimized structures are robust with respect to variations in geometric and material parameters as well as external loads it is therefore necessary to set up problem formulations in which such uncertain variations are accounted for. The author and co-workers have recently proposed to formulate TO under load-uncertainty as a two-person game in which the two players are the ”designer”, which controls the design variables, and ”nature”, which controls variables parametrizing the external loads. For this game a Nash-equilibrium is then sought which solves a standard TO problem to find the best design for a given load and an optimization problem – the ”load-problem” – to find the ”worst” load(s) for a given design (a problem of obvious interest also on its own). This work focuses on the load-problem – a non-convex problem with a relatively small number of variables and simple constraints – and investigates the practical possibility of solving it to global optimality using a simplicial branch-and-bound algorithm. The algorithm is tested on a topology optimized design to which a number of point loads are applied and allowed to vary in magnitude and direction in order to maximize a smooth approximation of the maximum von Mises stress. Numerical results show promise, but also that more work is needed to be able to solve large-scale (in terms of both the size of the structural model and the number of independent loads) instances of the load-problem to global optimality.*

1 INTRODUCTION

Topology optimization (TO) can be used to design load-carrying structures that are highly efficient with respect to *precisely* those conditions formulated mathematically in the optimization problem. Deviations, uncertain and inevitable in practise, from those conditions may cause optimized structures to behave erratically or even collapse. This leads to the idea of robust TO wherein uncertainties are accounted for explicitly in the optimization problem, either by assuming stochastic variations of problem parameters or using a *worst-case* approach [1]. Here we consider the latter approach and its use under load-uncertainty. This paper therefore concerns the problem of finding the ”worst” load(s) for a given functional f; i.e. the load(s) which maximize (or minimize) f. The setting, studied by the author and co-workers in [2, 3, 4], is a two-player Nash game formulation for robust TO wherein a ”designer” controls the design variables in $\boldsymbol{x}$ and ”nature” the load variables in $\boldsymbol{r}$. Assuming the same pay-off functional f for both players a Nash equilibrium $(\boldsymbol{x}^*, \boldsymbol{r}^*)$ is defined by

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}, \boldsymbol{r}^*) \tag{1a}$$

$$\boldsymbol{r}^* \in \arg\max_{\boldsymbol{r}\in\mathcal{T}} f(\boldsymbol{x}^*, \boldsymbol{r}) \tag{1b}$$

where $\mathcal{X} \subset \mathbb{R}^m$ and $\mathcal{T} \subset \mathbb{R}^d$ is the strategy set of respective player. Ideally (and suggested by the notation), $\boldsymbol{x}^*$ and $\boldsymbol{r}^*$ should be globally optimal solutions in their respective problem. Unfortunately the (standard) TO problem (1a) typically has a very large number of variables and solving it to global optimality is currently not a viable option. In problem (1b) – the ”load-problem” – however, the design is fixed and a useful parametrization of the load can often be obtained using a relatively small number of variables. Obtaining a globally optimal solution is also in one sense more important in (1b) than in (1a) as it ensures robustness of the optimized design; i.e., that there are no other loads realizable by the given parametrization that result in worse performance. For these reasons the focus here is the load-problem, for which globally optimal solutions are sought.

If f is the so-called *compliance* and the parametrization of the load is suitably chosen the load-problem is readily solved: the worst load can be obtained by solving an eigenvalue problem [5] or a generalized eigenvalue problem [6]. Here however, we let f be an ℓ_p-norm approximation of the maximum von Mises stress. In this case no ”closed-form” solution is, to the author's knowledge, available and the load-problem has to be treated as a non-convex, non-linear optimization problem (NLP). We propose to solve this problem to global optimality by a branch and bound algorithm rather than some type of heuristic method (such as gradient-based multistart or genetic algorithms). The obvious advantage of this choice is that we get a guarantee of the quality of the solution in the sense that the computed optimal value deviates at most a user-specified number ε from the globally optimal value.

Related work in solid mechanics includes [7] and [8] which describe the use of exact global optimization to evaluate a fatique criterion and the minimum limit load factor in worst case plastic analysis, respectively. Branch and bound has also been used for design of trusses under fixed loading conditions [9, 10].

In the following we describe the structural model and the parametrization of the external loads. We then define the ℓ_p-norm approximation of the maximum von Mises stress to be used as objective function in the load-problem. Thereafter a simplical branch and bound method is described (the method itself is rather standard [11, 12] and is described here mainly for the reader's convenience). Finally some numerical examples are described to give an idea of how the method works and performs.

2 THE MODEL AND ITS PARAMETRIZATION

We consider an elastic body undergoing small deformations. Using the finite element method we obtain the global equilibrium equation

$$\boldsymbol{K}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{r}) \tag{2}$$

for the nodal displacements in $\boldsymbol{u} \in \mathbb{R}^N$. Following the SIMP-approach to TO the global stiffness matrix

$$\boldsymbol{K}(\boldsymbol{x}) = \sum_{e=1}^{m} \rho_e(\boldsymbol{x})^q \boldsymbol{K}_e,$$

where the (relative) density-variables ρ_e depend on the design variables in $\boldsymbol{x}$ through a, in this case linear, filter [13], and $q > 1$. The matrices $\boldsymbol{K}_e$ are element stiffness matrices for material of unit (relative) density. The design $\boldsymbol{x}$ is assumed to be such that $\boldsymbol{K}(\boldsymbol{x})$ is positive definite.

The load vector in (2) is taken as

$$\boldsymbol{f}(\boldsymbol{r}) = \boldsymbol{f}_0 + \sum_{i=1}^{n} \boldsymbol{Q}_i^{\mathsf{T}} \boldsymbol{r}_i, \tag{3}$$

where $\boldsymbol{f}_0$ is a fixed vector, and $\boldsymbol{r}_i \in \mathbb{R}^s$, where s is the number of spatial dimensions, account for load variations during the optimization. The matrices $\boldsymbol{Q}_i$ can be used to specify which nodes are loaded and the maximum load variation in different directions. The vectors $\boldsymbol{r}_i$ are collected in $\boldsymbol{r} \in \mathcal{T} = \{\boldsymbol{r} \in \mathbb{R}^d \mid ||\boldsymbol{r}_i|| \leq 1, \ i = 1, \ldots, n\}$, where $d = sn$ and $|| \cdot ||$ is the Euclidean vector norm.

3 THE GLOBAL STRESS MEASURE

In this section we omit all dependencies on $\boldsymbol{x}$ as it is fixed to $\boldsymbol{x}^*$ in (1b).

Given the matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & -1/2 & 0 \\ -1/2 & 1 & 0 \\ 0 & 0 & 3 \end{pmatrix},$$

the von Mises stress at stress evaluation point a can be written as

$$\sigma_a^{vM}(\boldsymbol{r}) = \left[\boldsymbol{\sigma}_a(\boldsymbol{r})^{\mathsf{T}} \boldsymbol{A} \boldsymbol{\sigma}_a(\boldsymbol{r})\right]^{1/2}, \tag{4}$$

where the stress vector is, via (2), given by

$$\boldsymbol{\sigma}_a(\boldsymbol{r}) = \boldsymbol{E}_a \boldsymbol{B}_a \boldsymbol{u}(\boldsymbol{r}) = \boldsymbol{E}_a \boldsymbol{B}_a \boldsymbol{K}^{-1} \boldsymbol{f}(\boldsymbol{r}). \tag{5}$$

Here $\boldsymbol{B}_a$ is the (expanded) strain-displacement matrix evaluated at a and

$$\boldsymbol{E}_a = \rho_e^r \boldsymbol{E}_{0a},$$

where $r > 0$ is a constant, the index e indicates the finite element in which point a is found, and the symmetric, positive semi-definite matrix $\boldsymbol{E}_{0a}$ is the constitutive matrix for material of unit relative density.

Following a currently popular approach to TO with stress constraints/objectives [14, 15] we use as a global stress measure

$$\left(\sum_{a=1}^{S} \sigma_a^{vM}(\boldsymbol{r})^p\right)^{1/p}, \tag{6}$$

where $p > 0$. This is the ℓ_p-norm of the vector of von Mises stresses and it converges from above to $\max_{a\in\{1,\ldots,S\}} \sigma_a^{vM}(\boldsymbol{r})$ as p tends to infinity. Since for $p > 0$ the function $x \to x^{1/p}$ is monotone increasing on $[0, \infty)$ we omit the outer exponentiation in (6) and consider maximizing

$$\overline{\sigma}(\boldsymbol{r}) = \sum_{a=1}^{S} \sigma_a^{vM}(\boldsymbol{r})^p. \tag{7}$$

Since the design $\boldsymbol{x}$ is fixed, (7) can be evaluated very efficiently by a rewriting of (4). Let $\boldsymbol{Q}^{\mathsf{T}} = [\boldsymbol{Q}_1^{\mathsf{T}} \, \ldots \, \boldsymbol{Q}_2^{\mathsf{T}}] \in \mathbb{R}^{N\times d}$. Then, recalling (5) and (3),

$$\sigma_a^{vM}(\boldsymbol{r}) = \left[(\boldsymbol{f}_0 + \boldsymbol{Q}^{\mathsf{T}}\boldsymbol{r})^{\mathsf{T}}\boldsymbol{K}^{-1}\boldsymbol{B}_a^{\mathsf{T}}\boldsymbol{E}_a\boldsymbol{A}\boldsymbol{E}_a\boldsymbol{B}_a\boldsymbol{K}^{-1}(\boldsymbol{f}_0 + \boldsymbol{Q}^{\mathsf{T}}\boldsymbol{r})\right]^{1/2} =$$

$$\left[\boldsymbol{u}_0^{\mathsf{T}}\boldsymbol{W}_a\boldsymbol{u}_0 + \boldsymbol{r}^{\mathsf{T}}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{W}_a\boldsymbol{U}\boldsymbol{r} + 2\boldsymbol{u}_0^{\mathsf{T}}\boldsymbol{W}_a\boldsymbol{U}\boldsymbol{r}\right]^{1/2},$$

where $\boldsymbol{W}_a = \rho_e^r\boldsymbol{B}_a^{\mathsf{T}}\boldsymbol{E}_{0a}\boldsymbol{A}\boldsymbol{E}_{0a}\boldsymbol{B}_a$ is a very sparse matrix, and $\boldsymbol{u}_0 \in \mathbb{R}^N$ and $\boldsymbol{U} \in \mathbb{R}^{N\times d}$ solves the $sn + 1$ linear systems

$$\boldsymbol{K}[\boldsymbol{u}_0 \, \boldsymbol{U}] = [\boldsymbol{f}_0 \, \boldsymbol{Q}^{\mathsf{T}}].$$

These equations need only be solved once before the optimization process starts, and the von Mises stress is then given by

$$\sigma_a^{vM}(\boldsymbol{r}) = \left[v_a + 2\boldsymbol{v}_a^{\mathsf{T}}\boldsymbol{r} + \boldsymbol{r}^{\mathsf{T}}\boldsymbol{V}_a\boldsymbol{r}\right]^{1/2},$$

where $v_a = \boldsymbol{u}_0^{\mathsf{T}}\boldsymbol{W}_a\boldsymbol{u}_0$, $\boldsymbol{v}_a = \boldsymbol{U}^{\mathsf{T}}\boldsymbol{W}_a\boldsymbol{u}_0 \in \mathbb{R}^d$, and $\boldsymbol{V}_a = \boldsymbol{U}^{\mathsf{T}}\boldsymbol{W}_a\boldsymbol{U} \in \mathbb{R}^{d\times d}$.

4 THE LOAD-PROBLEM

In the special case when the fixed part of the load $\boldsymbol{f}_0 = \boldsymbol{0}$, (7) and (4) gives $\overline{\sigma}(\boldsymbol{r}) = \overline{\sigma}(-\boldsymbol{r})$. This implies (see Appendix A) that

$$\{\,\overline{\sigma}(\boldsymbol{r}) \mid \boldsymbol{r} \in \mathcal{T}\,\} = \{\,\overline{\sigma}(\boldsymbol{r}) \mid \boldsymbol{r} \in \mathcal{T},\, \boldsymbol{n}^{\mathsf{T}}\boldsymbol{r} \geq 0\,\}$$

for some vector $\boldsymbol{n}$. In view of this we now introduce

$$\mathcal{T}_+ = \{\,\boldsymbol{r} \in \mathbb{R}^d \mid ||\boldsymbol{r}_i|| \leq 1,\ i = 1, \ldots, n,\ \boldsymbol{n}^{\mathsf{T}}\boldsymbol{r} \geq 0\,\} \tag{8}$$

(the case when $\boldsymbol{f}_0 \neq \boldsymbol{0}$ is retrieved by letting $\boldsymbol{n} = \boldsymbol{0}$) and consider problem (1b), with $\sigma = -\overline{\sigma}$ in place of f and feasible set $\mathcal{T}_+$ instead of $\mathcal{T}$:

$$\min_{\boldsymbol{r}\in\mathcal{T}_+} \sigma(\boldsymbol{r}). \tag{9}$$

This problem is one of minimizing a smooth, concave function [2] over a compact, convex set. Existence of globally optimal solutions is thus guaranteed, and the problem is of a type much studied in the global optimization literature [12, 11].

4.1 A BRANCH-AND-BOUND ALGORITHM FOR THE LOAD-PROBLEM

We describe a branch-and-bound algorithm to solve (9) where the feasible set $\mathcal{T}_+$ is partitioned into simplices $\mathcal{S}_i$. A d-simplex in $\mathbb{R}^d$ is defined as the convex hull of $d+1$ affinely independent vertices $\boldsymbol{v}_0, \ldots, \boldsymbol{v}_d$, i.e.

$$\boldsymbol{r} \in \mathcal{S}_i \;\Leftrightarrow\; \boldsymbol{r} = \sum_{i=0}^{d} \lambda_i \boldsymbol{v}_i,\; \sum_{i=0}^{d} \lambda_i = 1,\; \lambda_i \geq 0,\; i = 0, \ldots, d. \tag{10}$$

To specify the algorithm we need to know: i) how to construct an initial simplex; ii) how to obtain upper and lower bounds on a simplex; iii) how to perform the partition of $\mathcal{T}_+$; and iv) how to check whether $\mathcal{T}_+ \cap \mathcal{S}_i = \emptyset$. These items are described next.

4.1.1 AN INITIAL SIMPLEX

Following [12, Section 3.5.3] we choose an initial simplex $\mathcal{S}_0 \supset \mathcal{T}_+$ as

$$\mathcal{S}_0 = \left\{ \boldsymbol{r} \in \mathbb{R}^d \mid r_i \geq \gamma_i, i = 1, \ldots, d,\; \sum_{i=1}^{d} r_i \leq \gamma \right\},$$

where $\gamma, \gamma_1, \ldots, \gamma_d$ are obtained by solving the convex problems

$$\gamma_i = \min_{\boldsymbol{r} \in \mathcal{T}_+} r_i, \quad i = 1, \ldots, d, \quad \text{and} \quad \gamma = \max_{\boldsymbol{r} \in \mathcal{T}_+} \sum_{i=1}^{d} r_i.$$

The vertices of $\mathcal{S}_0$ are then given by $\boldsymbol{v}_0 = (\gamma_1, \ldots, \gamma_d)^{\mathsf{T}}$ and

$$\boldsymbol{v}_i = (\gamma_1, \ldots, \gamma_{i-1}, \alpha_i, \gamma_{i+1}, \ldots, \gamma_d)^{\mathsf{T}}, \quad i = 1, \ldots, d,$$

where $\alpha_i = \gamma - \sum_{j \neq i} \gamma_j$.

4.1.2 UPPER BOUNDING

Assuming $\mathcal{T}_+ \cap \mathcal{S}_i \neq \emptyset$, an upper bound U_i is obtained on $\mathcal{T}_+ \cap \mathcal{S}_i$ by solving the problem

$$U_i = \min_{\boldsymbol{r} \in \mathcal{T}_+ \cap \mathcal{S}_i} \sigma(\boldsymbol{r}). \tag{11}$$

It is possible to treat this problem in two different ways: (i) compute the polytope representation of $\mathcal{S}_i$ [16, p. 33] and optimize directly in $\boldsymbol{r}$; or (ii) use the vertex representation (10) with the λ_i-s as variables. Here we choose the former strategy because of convenience in the implementation. For problems with more variables than the ones treated here, the fact that the polytope representation requires solution of some additional linear systems compared to the vertex representation may be one reason to prefer the latter.

To represent $\mathcal{S}_i$ as a polyhedron we follow [16, p. 33] and introduce a matrix

$$\boldsymbol{B}_i = [\boldsymbol{v}_1 - \boldsymbol{v}_0 \;\ldots\; \boldsymbol{v}_d - \boldsymbol{v}_0]^{-1} \in \mathbb{R}^{d \times d},$$

where the inverse is well-defined since $\boldsymbol{v}_0, \ldots, \boldsymbol{v}_d$ are affinely independent. Now [16, p. 33]

$$\mathcal{S}_i = \left\{ \boldsymbol{r} \in \mathbb{R}^d \mid \boldsymbol{B}_i \boldsymbol{r} \geq \boldsymbol{B}_i \boldsymbol{v}_0,\; \mathbf{1}^{\mathsf{T}} \boldsymbol{B}_i \boldsymbol{r} \leq \mathbf{1}^{\mathsf{T}} \boldsymbol{B}_i \boldsymbol{v}_0 + 1 \right\},$$

where $\mathbf{1}$ is a vector of ones.

4.1.3 LOWER BOUNDING

Since $\sigma(\boldsymbol{r})$ is concave, its convex envelope over $\mathcal{S}_i$ is [12, Theorem 1.22] the affine function $\ell(\boldsymbol{r}) = \boldsymbol{a}^\mathsf{T}\boldsymbol{r} + b$, where $\boldsymbol{a}$ and b comprise the unique solution to the linear system

$$\boldsymbol{a}^\mathsf{T}\boldsymbol{v}_i + b = \sigma(\boldsymbol{v}_i), \quad i = 0, \ldots, d.$$

Assuming $\mathcal{T}_+ \cap \mathcal{S}_i \neq \emptyset$, a lower bound L_i on $\mathcal{T}_+ \cap \mathcal{S}_i$ is obtained by solving

$$L_i = \min_{\boldsymbol{r} \in \mathcal{T}_+ \cap \mathcal{S}_i} \ell(\boldsymbol{r}). \tag{12}$$

Since the intersection of two convex sets is convex, this problem consists of minimizing a linear function over a convex set.

In this case, solutions to the lower bounding problem are feasible in the original problem (9), so, in order to obtain an upper bound U_i, simply evaluating σ at a solution to (12) is an alternative to solving (11)

4.1.4 SUBDIVISION OF SIMPLICES

Subdivision of a simplex $\mathcal{S}_i$ is done by bisection along one of its longest edges. Letting $\boldsymbol{v}_k$ and $\boldsymbol{v}_{k+1}$ be the end-points of this edge we take the bisection point as $\boldsymbol{w} = 0.5(\boldsymbol{v}_k + \boldsymbol{v}_{k+1})$ and thus obtain two new simplices $\mathcal{S}_{i1}$ and $\mathcal{S}_{i2}$ with vertices $(\boldsymbol{v}_0, \ldots, \boldsymbol{v}_k, \boldsymbol{w}, \boldsymbol{v}_{k+2}, \ldots, \boldsymbol{v}_d)$ and $(\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{k-1}, \boldsymbol{w}, \boldsymbol{v}_{k+1}, \ldots, \boldsymbol{v}_d)$.

4.1.5 INTERSECTION BETWEEN $\mathcal{T}_+$ AND $\mathcal{S}_i$

It could happen that the intersection $\mathcal{T}_+ \cap \mathcal{S}_i$ is empty, in which case $\mathcal{S}_i$ should be discarded from further consideration. Here we propose to check for an empty intersection in three steps[1]:

1. If any of the vertices of $\mathcal{S}_i$ lie in $\mathcal{T}_+$, then $\mathcal{T}_+ \cap \mathcal{S}_i \neq \emptyset$ and $\mathcal{S}_i$ should not be discarded.

2. Let $\boldsymbol{I}_j$ be a diagonal matrix with diagonal elements $2j-1$ and $2j$ set to one and the other to zero. If
$$\max_{j \in \{1,\ldots,n\}} \left[-\boldsymbol{v}_0^\mathsf{T}\boldsymbol{I}_j\boldsymbol{v}_0 + 2 \min_{i \in \{1,\ldots,d\}} \boldsymbol{v}_0^\mathsf{T}\boldsymbol{I}_j\boldsymbol{v}_i \right] > 1,$$
then $\mathcal{T}_+ \cap \mathcal{S}_i = \emptyset$ (see Appendix B) and $\mathcal{S}_i$ should be discarded.

3. If neither of Step 1 or 2 hold we proceed to solving the convex feasibility problem
$$\begin{aligned} z^* = &\min_{z \in \mathbb{R},\, \boldsymbol{r} \in \mathcal{S}_i} z \\ &\text{s. t.} \begin{cases} \boldsymbol{n}^\mathsf{T}\boldsymbol{r} \geq 0 \\ ||\boldsymbol{r}_i||^2 - 1 \leq z, \quad i = 1, \ldots, n. \end{cases} \end{aligned} \tag{13}$$
If $z^* \leq 0$ we conclude that $\mathcal{T}_+ \cap \mathcal{S}_i \neq \emptyset$, and $\mathcal{S}_i$ should not be discarded.

[1] Since the criteria in Step 2 is *sufficient* for an empty intersection, it is also possible to skip Step 3 altogether without losing candidate solutions to (9).

4.2 THE COMPLETE ALGORITHM

Let $\mathcal{A}$ denote a set of *problems*. A problem i in $\mathcal{A}$ is defined simply by the vertices of a simplex $\mathcal{S}_i$ and is associated with a lower bound L_i. The proposed algorithm for solving (9) to global optimality (to within a certain tolerance) can now be described as follows:

Algorithm 1

Choose $\varepsilon > 0$. Construct an initial simplex $\mathcal{S}_0$ as described in section 4.1.1 and add it to $\mathcal{A}$. Solve the lower bounding problem (12) and take the objective function value as the best lower bound L. Set the best upper bound U to ∞.

1. Select the problem, i, with the least lower bound and delete it from $\mathcal{A}$.
2. (Optional) Solve the upper bounding problem (11). If the optimal value U_i is smaller than U then set $U := U_i$.
3. Divide $\mathcal{S}_i$ into $\mathcal{S}_{i1}$ and $\mathcal{S}_{i2}$.
4. For $j = 1, 2$: If $\mathcal{S}_{ij} \cap \mathcal{T}_+ \neq \emptyset$, solve the lower bounding problem on $\mathcal{S}_{ij}$. If $L_{ij} < U$, then add problem ij to $\mathcal{A}$.
5. Set L to the least lower bound found in $\mathcal{A}$.
6. If $U - L < \varepsilon$, then STOP. Else goto Step 1.

Let σ^* denote the global minimum value of problem (9). Then according to Theorem 5.26 in [11] Algorithm 1 converges in a finite number of iterations to a point $\boldsymbol{r}^*$ such that $\sigma(\boldsymbol{r}^*) \leq \sigma^* + \varepsilon^2$.

5 NUMERICAL EXAMPLES

We consider solutions to some instances of (9) based on the design (i.e. $\boldsymbol{x}^*$) shown in the left plot of Fig. 1. This design was obtained in [2] by varying, under an upper bound on the total mass, $\boldsymbol{x}$ to minimize the ℓ_{24}-norm of the von Mises-stress vector and varying the load to maximize the same quantity. The L-shaped design domain is indicated by thin dashed lines. The finite element model consists of 6400 four-noded, bilinear elements, and stress is evaluated at the geometric center of each element. To see the effect of varying the design we also consider a problem where the design variables are set to one in the entire design domain. This design is denoted by ”s” in Tab. 1 below; the one seen in Fig. 1 by ”c”.

The load is given by (3) with $\boldsymbol{f}_0 = \boldsymbol{0}$, i.e., $\boldsymbol{f}(\boldsymbol{r}) = \sum_{i=1}^{n} \boldsymbol{Q}_i^{\mathsf{T}} \boldsymbol{r}_i$. Each $\boldsymbol{Q}_i$ consists here of zeros except for an identity matrix placed at the positions corresponding to the degrees of freedom of a loaded node; i.e. we consider point loads at various parts of the structure varying freely expect that their magnitude is limited to unity and that $\boldsymbol{n}^{\mathsf{T}}\boldsymbol{r} \geq 0$. Here we take $\boldsymbol{n} = (1 \ 0 \ \ldots \ 0)^{\mathsf{T}}$; i.e., we require that $r_1 \geq 0$. Stresses in a small patch of 3 by 4 elements around the point of application of each force are set to zero; c.f. [14, 15].

The right plot in Fig. 1 shows normalized maximum von Mises stress and its approximation using the ℓ_{24}-norm as functions of the angle θ defining the direction of the unit load applied at the tip of the beam in the left plot (here $n = 1$). Three local maxima are clearly discerned in the solid graph, illustrating the non-convexity of problem (9).

Algorithm 1 has been implemented in Matlab R2015b. The NLPs are solved by an interior-

[2] *Exactness in the limit*, as required by the cited theorem, follows from the Lipschitz continuity of σ and that ℓ interpolates σ at the vertices of each simplex. *Exhaustivness of the subdivision process* follows from Proposition 3.14 in [12].

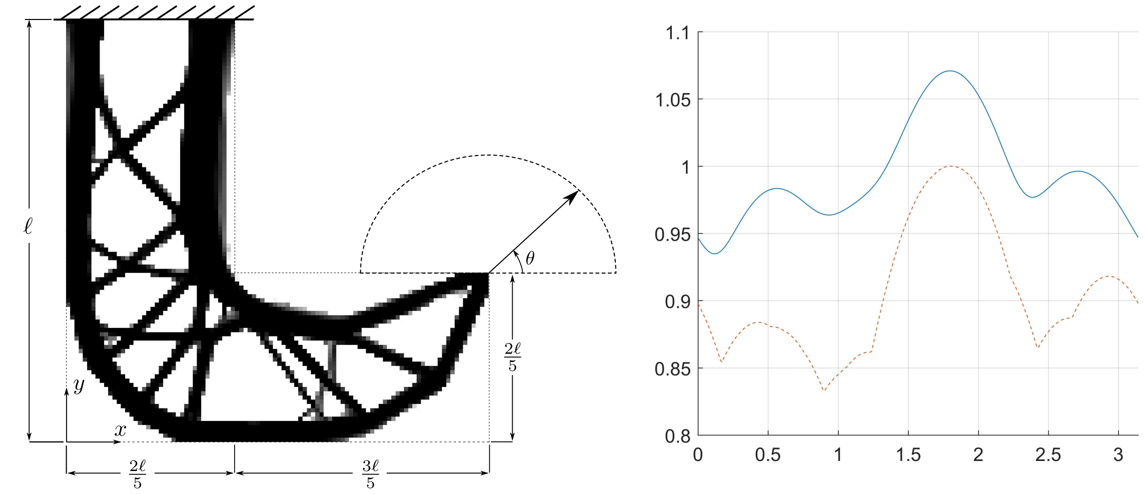

Figure 1: Left: Design with boundary conditions and loading. $\ell = 100$ [mm]. A point load is applied at the tip of the beam and the arrowhead is allowed to vary within the half-circle delimited by the dashed line. Right: Normalized maximum von Mises stress (dashed line) and its ℓ_{24}-norm-approximation (solid line) as functions of the angle θ in the left plot.

point algorithm available via the function `fmincon` in the Matlab Optimization Toolbox. The global stress measure (7) and its gradient are implement as C-functions interfaced to Matlab via the MEX API. Exact first-order derivatives are used for all problems. A BFGS approximation of the Hessian of the Lagrangian is used to solve (9), whereas exact Hessians are used for the other problems. In the model we use $q = 3$, $r = 0.5$; filter radius 3 [mm]; an isotropic material with Young's modulus 1000 [N/mm^2] and Poisson's ratio 0.3; and domain thickness 1 [mm]. The stopping tolerance in Algorithm 1 is set to $\varepsilon = 10^{-3}$. Stopping tolerances for `fmincon` are set to `TolX` $= 10^{-12}$, `TolCon` $= 10^{-10}$ and `TolFun` $= 10^{-10}$. The upper bounding problem in Step 2 of Algorithm 1 is solved every 50:th iteration. All computations are carried out by an Intel Core I7-4712MQ.

Table 1 shows some numerical results. In all cases the NLPs in Algorithm 1 were, after appropriate scaling, solved by `fmincon` using between 10 and 30 iterations for problems (12) and (13), and occasionally up to around 100 for problem (11). For the cases $n = 1$, $d = 2$ in the table, the total CPU time is on the order of 10 seconds and the number of iterations of Algorithm 1 around 200. For the three cases with $n = 2$, $d = 4$, the number of iterations jump to the order of 10000 and CPU times to between 700 and 1200 [s]. The absolute gap $U - L$ as a function of the number of iterations for the case in row three is shown in Fig. 2. Changing the design from the one seen in Fig. 1 to the simpler design "s" in the forth row of Tab. 1 yields a slight increase in computational time compared to the third row. For the cases $n = 3$, $d = 6$, the iteration count increased further. The total time increases with increasing p, with $p = 24$ resulting in around 38 hours of CPU time.

6 CONCLUSIONS

A branch and bound method for finding loads that maximize a given functional, in this case an ℓ_p-norm approximation of the maximum von Mises stress, has been described. The proposed method is quite general and can be applied to a wide range of functionals and for many different

n	d	p	des.	load position	iter.	f. eval.	low. bnd.	$\vert\mathcal{A}\vert$	CPU [s]
1	2	24	c	(100,40)	191	1129	343	17	14
1	2	24	c	(40,0)	161	1227	298	13	21
2	4	24	c	(100,40), (0,40)	13761	$1.3{\cdot}10^5$	$2.4{\cdot}10^4$	877	875
2	4	24	s	(100,40), (0,40)	20001	$1.8{\cdot}10^5$	$3.5{\cdot}10^4$	820	1197
2	4	24	c	(40,0), (0,40)	10631	$1{\cdot}10^5$	$1.8{\cdot}10^4$	668	710
3	6	4	c	(100,40), (0,40), (40,0)	$8.7{\cdot}10^4$	$1.2{\cdot}10^6$	$1.6{\cdot}10^5$	5913	6912
3	6	8	c	(100,40), (0,40), (40,0)	$3.3{\cdot}10^5$	$4.2{\cdot}10^6$	$5.7{\cdot}10^5$	15835	24418
3	6	24	c	(100,40), (0,40), (40,0)	$2{\cdot}10^6$	$25{\cdot}10^6$	$3.4{\cdot}10^6$	97540	139435

Table 1: Numerical results. Columns show number of individual loads; number of variables; ℓ_p-norm exponent; type of design; point of application of loads in the coordinate system shown in Fig. 1; number of iterations of Algorithm 1; number of evaluations of (7); number of times the lower bounding problem (12) has been solved; maximum number of active problems; and total time

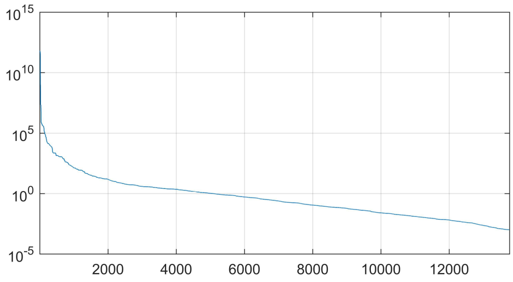

Figure 2: Absolute gap $U - L$ versus number of iterations in Algorithm 1 for the case $n = 2$, $d = 4$ described in the third row of Tab. 1.

loading parametrizations. A test problem based on a topology optimized design showed that the approach is promising, but, although further optimization of the code might bring down computational times somewhat, the large number of iterations observed indicate that more work is needed to be able to solve large-scale (in terms of both the size of the finite element model and the number of independent loads) instances of the load-problem to global optimality. Finally we remark that although the load-problem is here described as part of a Nash game for robust TO it is obviously very relevant on its own.

ACKNOWLEDGEMENTS

This research was supported in part by the Swedish Foundation for Strategic Research, Grant No. AM13-0029. Thanks to Associate Professor Johan Löfberg for useful comments on an early draft.

References

[1] A Ben-Tal, L El Ghaoui, and AS Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

[2] E Holmberg, C-J Thore, and A Klarbring. A game theory approach for robust topology optimization with uncertain loading. *Under review*, 2015.

[3] E Holmberg. *Topology optimization considering stress, fatique and load uncertainities*. PhD thesis, Linköping University, 2015.

[4] E Holmberg, C-J Thore, and A Klarbring. Game theory approach for topology optimization with uncertain loads. In *Svenska Mekanikdagar 10-12 Juni, Linköping*, 2015.

[5] E Holmberg, C-J Thore, and A Klarbring. Worst-case topology optimization of self-weight loaded structures using semi-definite programming. *Structural and Multidisciplinary Optimization*, 52(5):915–928, 2015.

[6] C-J Thore, E Holmberg, and A Klarbring. Large-scale robust topology optimization under load-uncertainty. In *11th World Congress on Structural and Multidisciplinary Optimisation*, 2015.

[7] H Svärd. A branch and bound algorithm for evaluation of the Findley fatigue criterion. *International Journal of Fatigue*, 73:27–38, 2015.

[8] Y Kanno and I Takewaki. Worst case plastic limit analysis of trusses under uncertain loads via mixed 0-1 programming. *Journal of Mechanics of Materials and Structures*, 2(2):245–273, 2007.

[9] M Stolpe. Global optimization of minimum weight truss topology problems with stress, displacement, and local buckling constraints using branch-and-bound. *International Journal for Numerical Methods in Engineering*, 61:1270–1309, 2004.

[10] M Stolpe and A Kawamoto. Design of planar articulated mechanisms using branch and bound. *Mathematical Programming*, 103:357–397, 2005.

[11] M Locatelli and F Schoen. *Global Optimization – Theory, Algorithms and Applications*. SIAM, 2013.

[12] R Horst, PM Pardalos, and N Van Thoai. *Introduction to global optimization*. Kluwer Academic Publishers, 1995.

[13] TE Bruns and DA Tortorelli. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 190(26):3443–3459, 2001.

[14] E Holmberg, B Torstenfelt, and A Klarbring. Stress constrained topology optimization. *Structural and Multidisciplinary Optimization*, 48(1):33–47, 2013.

[15] DM De Leon, J Alexandersen, JS Fonseca, and O Sigmund. Stress-constrained topology optimization for compliant mechanism design. *Structural and Multidisciplinary Optimization*, 52(5):929–943, 2015.

[16] S Boyd and L Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[17] U Raber. A simplicial branch-and-bound method for solving nonconvex all-quadratic programs. *Journal of Global Optimization*, 13(4):417–432, 1998.

[18] N Andréasson, A Evgrafov, and M Patriksson. *An Introduction to Continuous Optimization*. Studentlitteratur, 2005.

A IMAGES OF $\mathcal{T}$ AND $\mathcal{T}_+$

Proposition. Let $g : \mathbb{R}^d \to \mathbb{R}$ satisfy $g(\boldsymbol{r}) = g(-\boldsymbol{r})$ for all $\boldsymbol{r} \in \mathcal{T} = \{\boldsymbol{r} \in \mathbb{R}^d \mid ||\boldsymbol{r}_i|| \leq 1,\ i = 1, \ldots, n\}$. Then

$$\{\, g(\boldsymbol{r}) \,|\, \boldsymbol{r} \in \mathcal{T} \,\} = \{\, g(\boldsymbol{r}) \,|\, \boldsymbol{r} \in \mathcal{T}_+ \,\},$$

where $\mathcal{T}_+ = \{\, \boldsymbol{r} \in \mathbb{R}^d \mid ||\boldsymbol{r}_i|| \leq 1,\ i = 1, \ldots, n,\ \boldsymbol{n}^{\mathsf{T}}\boldsymbol{r} \geq 0 \,\}$ for some vector $\boldsymbol{n} \in \mathbb{R}^d$.

Proof. We have

$$\{\, g(\boldsymbol{r}) \,|\, \boldsymbol{r} \in \mathcal{T} \,\} = \{\, g(\boldsymbol{r}) \,|\, \boldsymbol{r} \in \mathcal{T},\ \boldsymbol{n}^{\mathsf{T}}\boldsymbol{r} \geq 0 \,\} \cup \{\, g(\boldsymbol{r}) \,|\, \boldsymbol{r} \in \mathcal{T},\ \boldsymbol{n}^{\mathsf{T}}\boldsymbol{r} \leq 0 \,\} = U_+ \cup U_- \quad (14)$$

Since $||\boldsymbol{r}_i|| \leq 1$ and $\boldsymbol{n}^{\mathsf{T}}\boldsymbol{r} \leq 0$ if and only if if and only if $|| - \boldsymbol{r}_i|| \leq 1$ and $\boldsymbol{n}^{\mathsf{T}}(-\boldsymbol{r}) \geq 0$ we can write

$$U_- = \{\, g(\boldsymbol{r}) \mid -\boldsymbol{r} \in \mathcal{T},\ \boldsymbol{n}^{\mathsf{T}}(-\boldsymbol{r}) \geq 0 \,\} = \{\, g(-\boldsymbol{r}) \mid -\boldsymbol{r} \in \mathcal{T},\ \boldsymbol{n}^{\mathsf{T}}(-\boldsymbol{r}) \geq 0 \,\}.$$

Now defining $\boldsymbol{s} \equiv -\boldsymbol{r}$ we see that $U_- = U_+$, so that, going back to (14),

$$\{\, g(\boldsymbol{r}) \,|\, \boldsymbol{r} \in \mathcal{T} \,\} = U_+ \cup U_- = U_+ \cup U_+ = U_+.$$

□

B A SUFFICIENT CONDITION FOR $\mathcal{T}_+ \cap \mathcal{S}_i = \emptyset$

Proposition. Let $\boldsymbol{I}_j$ be a diagonal matrix with ones in diagonal elements $2j - 1$ and $2j$ and zeros otherwise. If

$$\max_{j \in \{1,\ldots,n\}} \left[-\boldsymbol{v}_0^{\mathsf{T}}\boldsymbol{I}_j\boldsymbol{v}_0 + 2 \min_{i \in \{1,\ldots,d\}} \boldsymbol{v}_0^{\mathsf{T}}\boldsymbol{I}_j\boldsymbol{v}_i \right] > 1, \quad (15)$$

then $\mathcal{T}_+ \cap \mathcal{S}_i = \emptyset$.

(A version of this criteria was originally proposed by Raber [17].)

Proof. Let $\boldsymbol{r}$ denote a vector from the origin to an arbitrary point in $\mathcal{S}_i$. Then

$$\boldsymbol{r}^{\mathsf{T}}\boldsymbol{I}_j\boldsymbol{r} = (\boldsymbol{v}_0 + (\boldsymbol{r} - \boldsymbol{v}_0))^{\mathsf{T}}\boldsymbol{I}_j(\boldsymbol{v}_0 + (\boldsymbol{r} - \boldsymbol{v}_0)) = \boldsymbol{v}_0^{\mathsf{T}}\boldsymbol{I}_j\boldsymbol{v}_0 + 2\boldsymbol{v}_0^{\mathsf{T}}\boldsymbol{I}_j(\boldsymbol{r} - \boldsymbol{v}_0) + (\boldsymbol{r} - \boldsymbol{v}_0)^{\mathsf{T}}\boldsymbol{I}_j(\boldsymbol{r} - \boldsymbol{v}_0) \geq \boldsymbol{v}_0^{\mathsf{T}}\boldsymbol{I}_j\boldsymbol{v}_0 + 2\boldsymbol{v}_0^{\mathsf{T}}\boldsymbol{I}_j(\boldsymbol{r} - \boldsymbol{v}_0), \quad (16)$$

where the last inequality follows from $\boldsymbol{I}_j$ being positive semi-definite. We now write $\boldsymbol{r} = \boldsymbol{v}_0 + \sum_{i=1}^{d} \lambda_i(\boldsymbol{v}_i - \boldsymbol{v}_0)$, where $\lambda_i \geq 0$, $i = 1, \ldots, d$ and $\sum_{i=1}^{d} \lambda_i \leq 1$. Substitution in (16) gives

$$\boldsymbol{r}^{\mathsf{T}}\boldsymbol{I}_j\boldsymbol{r} \geq \boldsymbol{v}_0^{\mathsf{T}}\boldsymbol{I}_j\boldsymbol{v}_0 + 2\boldsymbol{v}_0^{\mathsf{T}}\left(\sum_{i=1}^{d} \lambda_i\boldsymbol{I}_j(\boldsymbol{v}_i - \boldsymbol{v}_0)\right) \geq \boldsymbol{v}_0^{\mathsf{T}}\boldsymbol{I}_j\boldsymbol{v}_0 + \min_{\lambda_i \geq 0,\, i=1,\ldots,d,\, \sum_{i=1}^{d}\lambda_i \leq 1} 2\boldsymbol{v}_0^{\mathsf{T}}\left(\sum_{i=1}^{d} \lambda_i\boldsymbol{I}_j(\boldsymbol{v}_i - \boldsymbol{v}_0)\right) \geq \boldsymbol{v}_0^{\mathsf{T}}\boldsymbol{I}_j\boldsymbol{v}_0 + \min_{i \in \{1,\ldots,d\}} 2\boldsymbol{v}_0^{\mathsf{T}}\boldsymbol{I}_j(\boldsymbol{v}_i - \boldsymbol{v}_0). \quad (17)$$

The last inequality follows from the fact that for a linear program with a bounded feasible set, at least one extreme point of this set will be a solution [18, Theorem 8.10]; i.e, at least one of the vertices of $\mathcal{S}_i$ will be a solution. The last inequality in (17) shows that $\boldsymbol{v}_0^\mathsf{T}\boldsymbol{I}_j\boldsymbol{v}_0 + \min_{i\in\{1,\ldots,d\}} 2\boldsymbol{v}_0^\mathsf{T}\boldsymbol{I}_j(\boldsymbol{v}_i - \boldsymbol{v}_0) > 1$ implies $\boldsymbol{r}^\mathsf{T}\boldsymbol{I}_j\boldsymbol{r} > 1$ and thus, since $\boldsymbol{r} \in \mathcal{S}_i$ is arbitrary, that $\mathcal{T}_+ \cap \mathcal{S}_i = \emptyset$. $\square$

Remark. The smaller the simplex, the smaller is the term $(\boldsymbol{r} - \boldsymbol{v}_0)^\mathsf{T}\boldsymbol{I}_j(\boldsymbol{r} - \boldsymbol{v}_0)$ in (16) and the better is the accuracy of the criterion (15).